

解説



最小二乗法の新しいアルゴリズム†

小柳 義夫††

1. はじめに

FORTRAN の入門書では、数学で用いられる代数式と同じ形で算術演算が記述できることが強調されている。これは初心者に無用の恐れを抱かせない配慮なのであろうが、その段階に留ったままの人が少ないのは残念である。数式通りプログラムを書いてはいけない——これは数値計算のいわば鉄則である。

数値計算は有限桁の演算であるため、数学的に正しい式だからといってそのまま成り立つわけではない。結合則 $(a+b)+c=a+(b+c)$ や分配則 $a(b+c)=ab+ac$ も満たされない。計算機を用いて数値計算を行うには、できるだけ誤差が小さく、かつ計算速度の速いアルゴリズムを工夫する必要がある。とくに、電子計算機はブラックボックスであり、途中経過を目で見るができないので、手計算や手まわし計算機の時代のアルゴリズムとはまた別の配慮が必要になる。本稿では線形および非線形の最小二乗法を材料に、数値計算のアルゴリズムについて考察する。

最小二乗法は、C. F. Gauss が 1795 年に観測データの整理のため考察したのが最初であり、今でもあらゆる分野のデータ解析のために用いられている。データ解析は、データの検討、モデルの吟味などを含む複雑な過程であるが、その中心となるのが最小二乗法によるあてはめである。最近のように、精密なデータが多量に得られるようになると、これを説明するために多数の変数パラメータを含む複雑なモデルを扱う必要が生じる。超高速の電子計算機でも総計数百時間の CPU 時間を要するような問題も決して稀ではなく、アルゴリズムの良し悪しは決定的な影響を与える。小規模な問題でも、2~3 個のパラメータを含む指数関数モデルによって 2~30 個のデータを解析するというような場合、悪いプログラムではすぐ発散してしまう。まし

て大きな問題では細心の注意が必要とされる。

2 章では、線形最小二乗法について、Gauss 以来用いられてきた正規方程式を丸め誤差の分析によって批判し、QR 分解や特異値分解に基づく方法について述べる。3 章ではその具体的なアルゴリズムを示し、数値例について考察する。4 章では非線形最小二乗法、とくに Gauss-Newton 系の方法について解説し、最近脚光を浴びてきた Newton 系の方法と比較する。

2. 線形最小二乗法

2.1 数値計算の誤差

計算機の中での数値計算の精度の目安として、計算機イプシロン (machine epsilon) ϵ_M という指標が用いられる。 ϵ_M は $1+\epsilon > 1$ を満たす最小の正数 ϵ のことである。数学的には ϵ がいくら小さくてもこの式は成立するが、計算機の中の有限桁の表現では、 ϵ があまり小さくなると $1+\epsilon$ と 1 とは区別できなくなる。大型機で多く用いられる 4 バイト (32 ビット) の 16 進浮動小数点数 (FORTRAN の単精度実数) では $\epsilon_M = 16^{-6} = 0.954 \times 10^{-6}$ であり、8 バイトの倍精度実数では $\epsilon_M = 16^{-13} = 2.22 \times 10^{-16}$ である。計算機の単精度実数演算の精度は多くの電卓にも劣るほどである。このことは、数値計算のプログラミングに細心の注意が必要であることを示す。

浮動小数で表現できる範囲の任意の実数 x を、これに一番近い浮動小数 x_R で表現すると、

$$x_R = (1+\delta)x \quad \text{ただし} \quad |\delta| \leq \frac{1}{2}\epsilon_M \quad (2.1)$$

が成立する。また 2 個の浮動小数 x, y の加減乗除演算 (\circ で示す) の結果を浮動小数で表現したものを $f(x \circ y)$ とすれば、

$$f(x \circ y) = x \circ y (1+\delta) \quad \text{ただし} \quad |\delta| < \epsilon_M \quad (2.2)$$

が成立する (演算結果が切り捨てられる場合)。このように ϵ_M は、有限桁のために生じる誤差の指標とみることができる¹⁾。ただしこれは固定長仮数部表現の場合であって、最近提唱されている可変長指数部表現²⁾の場合の誤差解析はより複雑である。

† New Algorithms in Least Squares Method by Yoshio OYANAGI (Institute of Information Sciences, University of Tsukuba).

†† 筑波大学電子情報工学系

2.2 行列と条件数

一般の $(n \times m)$ 長方形行列 A によって, m 元ベクトル x は, n 元ベクトル $y = Ax$ に写像される. $n \geq m$ であつ A がフルランクの場合に, x の微変化小 δx が y に及ぼす変化 $\delta y = A\delta x$ は, ノルムの比で評価すると,

$$\frac{\|\delta y\|}{\|y\|} \frac{\|\delta x\|}{\|x\|} = \frac{\|A\delta x\|}{\|\delta x\|} \frac{\|Ax\|}{\|x\|} \leq \max_{|v|=1} \|Av\| / \min_{|v|=1} \|Av\| \quad (2.3)$$

となる. 上式の右辺を行列 A の条件数 (condition number) $\kappa(A)$ とよび, 行列 A を係数とする連立方程式の条件の悪さを表す. ノルムとしては何をとってもよいが, 以下ではユークリッド・ノルムを主に考えることにする.

n 元連立方程式

$$Ax = b \quad (2.4)$$

を解くことを考える. A は $(n \times n)$ 正則行列とする. この方程式をある解法で解いたところ, 丸め誤差のために右辺に誤差が生じ

$$A(x + \delta x) = b + \delta b \quad (2.5)$$

の解を求めたことになったとする. δb のために生じた x の誤差 δx については

$$\frac{\|\delta x\|}{\|x\|} \leq \kappa(A) \frac{\|\delta b\|}{\|b\|} \quad (2.6)$$

によって評価できる³⁾. すなわち解 x の相対誤差は, 右辺 b の相対誤差の $\kappa(A)$ 倍を越えない. 同様に, 係数行列 A に生じた誤差 δA による x の誤差 δx も

$$\frac{\|\delta x\|}{\|x + \delta x\|} \leq \kappa(A) \frac{\|\delta A\|}{\|A\|} \quad (2.7)$$

によって上限が与えられる. ただし, 行列 A のノルムは, $\|A\| = \max\{\|Ax\|/\|x\|\}$ によって定義される. 条件数 $\kappa(A)$ が ϵ_m^{-1} と同程度もしくはそれ以上の場合に, 行列 A は「条件が悪い (ill-conditioned)」という. このような場合の x の有効数字が 1 桁もなくなってしまう危険があるからである. b における 1 程度の誤差が, x に 10^9 の狂いを生じさせることも, 珍しくない.

ただしこの評価は上限であつて,

$$\begin{pmatrix} 1 & 0 \\ 0 & 10^{-6} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad (2.8)$$

のような場合には, 係数行列の条件数は 10^6 であるにもかかわらず, x_1, x_2 は有効数字一杯まで正しく求めることができる. 行列をスケーリングすれば, 条件数を 1 まで下げることができる. スケーリングによって

計算が安定化されることはよくあるが, 最適なスケーリングを求める一般的なアルゴリズムは知られていない.

2.3 最小二乗法

未知数の数 m より, 方程式の数 n の方が大きい場合の方程式

$$Ax = y \quad (2.9)$$

を優決定系といい, 一般的には上式を満たす x は存在しない. 条件を少し緩めて, 上式の左辺と右辺の差ができるだけ小さくなるような x を求めることを考える. 差の大きさの規準として, ユークリッド・ノルム (の二乗)

$$S(x) = \|y - Ax\|^2 = \sum_{i=1}^n (y_i - \sum_{j=1}^m A_{ij}x_j)^2 \quad (2.10)$$

をとれば, この問題は最小二乗法となる.

データ解析の場合は, 測定値 y_i (測定誤差の分散 σ_i^2) に対して, 未知パラメータ x_j に関して線形なモデル

$$\sum_{j=1}^m A_{ij}x_j + y_{0i} \quad (2.11)$$

を最小二乗法であてはめるには,

$$S(x) = \sum_{i=1}^n \left[y_i - \left(y_{0i} + \sum_{j=1}^m A_{ij}x_j \right) \right]^2 / \sigma_i^2 \quad (2.12)$$

を最小にするわけであるから, 適当に変数変換すれば式(2.10)に帰着する.

式(2.10)の最小値を求めるには, x に関する偏微分を 0 とおき,

$$A^T Ax = A^T y \quad (2.13)$$

を解けばよい. 上式は正規方程式とよばれ, Gauss 以来, 線形最小二乗法の標準的な解法として永く使われている. 大学の計算センタのライブラリの利用統計では, 連立一次方程式の解法がいつも上位にランクされているが, そのかなりの部分は正規方程式を解いているのではないかと推察される. 中には, 逆行列サブルーチンを使って, $x = (A^T A)^{-1} A^T y$ を数式通り計算している人さえ見うけられる.

ところが,

$$\kappa(A^T A) = [\kappa(A)]^2 \quad (2.14)$$

であるから, 正規方程式の条件数はかなり悪くなる危険がある. たとえば

$$A = \begin{pmatrix} 1 & 1 \\ \delta & 0 \\ 0 & \delta \end{pmatrix} \quad (2.15)$$

とすると $\kappa(A) \sim \delta^{-1}$ であるが,

$$A^T A = \begin{pmatrix} 1+\delta^2 & 1 \\ 1 & 1+\delta^2 \end{pmatrix} \quad (2.16)$$

となり, $\delta^2 < \epsilon_M$ であれば, $A^T A$ は特異行列と判定されてしまう。正規方程式は, 条件数をわざわざ悪くして解くことになる。もし $\kappa(A)$ のオーダの条件数の線形演算で式(2.13)を解くことができれば, 精度が大幅に向上することが期待される。

2.4 特異値分解

一つの方法は, 行列 A の特異値分解

$$A = U \Sigma V^T \quad (2.17)$$

U は $n \times n$, V は $m \times m$ の直交行列

Σ は対角成分のみをもつ $n \times m$ 長方形行列を用いることである^{3), 4)}。正規方程式(2.13)に上式を代入すれば,

$$V \Sigma^T \Sigma V^T x = V \Sigma^T U^T y, \quad (2.18)$$

したがって解 x は

$$x = V \Sigma^+ U^T y \quad (2.19)$$

となる。ここで Σ^+ は $m \times n$ の長方形対角行列であり, 対角成分は Σ の対応する対角成分の逆数とする。ここで, $\kappa(U) = \kappa(V) = 1$, $\kappa(\Sigma^+) = \kappa(\Sigma) = \kappa(A)$ であるから, 一応オーダ $\kappa(A)$ の条件数の式になっている。

この方法の決定的な問題点は, 特異値分解の計算に大変手間と時間がかかり, したがって誤差が入りやすいことである。さらに, ベクトル x の各成分は, 通常, 次元もスケールも異なるので, 直交変換であることの利点は必ずしもない。数値実験の結果も, 正規方程式よりはよいが, 次に述べる QR 分解による最小二乗法には及ばない。

2.5 QR 分解による最小二乗法

長方形行列 A の QR 分解⁴⁾ とは

$$A = QR \quad (2.20)$$

という分解で, Q は列ベクトルが正規直交系をなす $n \times m$ 長方形行列 (すなわち $Q^T Q = I$), R は $m \times m$ の上三角行列である。このような分解ができれば, 正規方程式(2.13)は

$$R^T R x = R^T Q^T y \quad (2.21)$$

となり, A がフルランクなら R は正則で, 上式は

$$R x = Q^T y \quad (2.22)$$

に帰着される。 R は上三角であるから, 上式は後退代入だけで解くことができる。

丸め誤差により, y に誤差 δy が生じたとすると, これによる x への影響は, $\kappa(Q) = 1$, $\kappa(R) = \kappa(A)$ を用いて

$$\frac{\|\delta x\|}{\|x\|} \leq \kappa(A) \frac{\|\delta y\|}{\|y\|} \quad (2.23)$$

となり, $\kappa(A)$ の1乗でおさえられる。他方 A に生じた誤差の影響の評価はかなり複雑である。一次方程式の場合と異なり, 誤差 δA が A の列ベクトルの張る空間 $\mathcal{R}(A)$ の外にずれる可能性があるからである。結果だけ示すと,

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\kappa(A)}{\sqrt{1-\alpha}} \left[1 + \frac{\kappa(A)}{\sqrt{1-\alpha}} \frac{\|y - Ax\|}{\|A\| \|x\|} \right] \frac{\|\delta A\|}{\|A\|} \quad (2.24)$$

ただし, $\alpha = (\sqrt{2} + 1) \|R^{-1}\| \|\delta A\| < 1$

となる⁴⁾。すなわち, 残差 $y - Ax$ が小さい場合にはほぼ $\kappa(A)$ に比例するが, 残差が大きい場合には $\kappa(A)^2$ に比例する成分をもつことがわかる。

Q の列ベクトルは $\mathcal{R}(A)$ の直交基底であり, R はこの基底に関する A の列ベクトルの各成分を与える。 $Q^T y$ は, y の $\mathcal{R}(A)$ への射影に対応している。大雑把に言って, パラメータ側の情報は R に, データ側の情報は Q に含まれている。たとえば推定したパラメータ \hat{x} の誤差行列 $\Sigma_{\hat{x}}$ は

$$\Sigma_{\hat{x}} = (A^T A)^{-1} = R^{-1} (R^{-1})^T \quad (2.25)$$

によって与えられる。 R が与えられれば, 上三角のスペースだけで R^{-1} を求めることができ, $\Sigma_{\hat{x}}$ の計算も容易である。他方, データに対する推定理論値 $\hat{\theta} = A \hat{x}$ の誤差行列 $\Sigma_{\hat{\theta}}$ は, σ_i を単位として

$$\Sigma_{\hat{\theta}} \sim A (A^T A)^{-1} A^T = Q Q^T \quad (2.26)$$

によって与えられる。すなわち, Q の i 番目の行ベクトルのノルムは, あてはめによって得た理論値の誤差 σ_{θ_i} と, 測定誤差 σ_i との比 (≤ 1) を与える。このように, QR 分解を用いることにより, 線形最小二乗法とそれに関連した統計計算を簡単に遂行することができる。

3. 数値計算としての最小二乗法

QR 分解は, 連立一次方程式のための LU 分解等と同じく, 有限回の手続きで実行することができる。ここでは, 行列 A を列単位に処理する Householder 法および Gram-Schmidt 法について解説するが, スパースな大行列に対しては, Givens 変換を用いる方法もある。

3.1 Householder 法

行列の固有値問題等に用いられる Householder 変換行列

$$P = I - 2ww^T \quad \text{ただし} \quad \|w\| = 1 \quad (3.1)$$

を用いて, QR 分解が実行できる⁴⁾. すなわち, m 回の変換により

$$A = P^{(1)}P^{(2)}\dots P^{(m)} \begin{pmatrix} R \\ 0 \end{pmatrix} = (QQ^T) \begin{pmatrix} R \\ 0 \end{pmatrix} = QR \quad (3.2)$$

とすることができる. ただし R は $m \times m$ の上三角行列であり, $P^{(1)}\dots P^{(m)}$ で表される $n \times n$ 直交行列を, $n \times m$ 行列 Q と, $n \times (n-m)$ 行列 Q' とに分解した. Q' は 0 に掛けるので消えてしまう.

具体的には $P^{(j)}$ を次のように構成する. $P^{(j-1)}\dots P^{(1)}A$ の第 j 列を $(a_{1j}, a_{2j}, \dots, a_{nj})^T$ とすると,

$$b = \left(\sum_{i=j}^n a_{ij}^2 \right)^{1/2} \quad (3.3)$$

$$w' = (0, 0, \dots, a_{1j} \pm b, a_{j+1,j}, \dots, a_{nj})^T \quad (3.4)$$

$$P = I - 2w'w'^T / \|w'\|^2 \quad (3.5)$$

により第 j 列は

$$P(a_{1j}, a_{2j}, \dots, a_{nj})^T = (a_{1j}, \dots, a_{j-1,j}, b, 0, \dots, 0)^T \quad (3.6)$$

のように変換される. 式 (3.4) の複号はどちらでもよいが, 桁落ちを避けるために a_{1j} と同符号にとる. このような変換を m 回施すことにより, A は上三角行列に変換される.

数値計算のアルゴリズムとしては, $n \times n$ 行列 P の形ではなく, w' のまま記憶し, P をベクトル y に掛ける場合は,

$$Py = y - 2w'(w'^T y) / \|w'\|^2 \quad (3.7)$$

の形で計算する. したがって, Householder 法の場合は, Q の具体的な形はどこにも記憶する必要はない. ただし σ_{q_i} を計算するには余分な計算が必要になる.

3.2 修正 Gram-Schmidt 法

m 個の一次独立なベクトル a_1, a_2, \dots, a_m から, 正規直交基底 q_1, q_2, \dots, q_m を構成する方法として, Gram-Schmidt の直交化という方法が知られている. これをそのまま A の QR 分解に応用したのが古典的 Gram-Schmidt 法であるが, Householder 法に比べて精度が少し劣る. これに対して Björk⁵⁾ は, 次のような改良を提案した.

あるベクトル a を, 正規直交系 (q_1, q_2, \dots, q_m) の成分に分解する際, 普通は

$$\text{for } i=1 \text{ to } m \text{ do} \\ a_i = q_i^T \cdot a \quad (3.8)$$

によって計算する. 他方一回ごとに対応する成分を a から引き去り

$$a^{(0)} = a \\ \text{for } i=1 \text{ to } m \text{ do}$$

$$a_i = q_i^T \cdot a^{(i-1)}, \quad a^{(i)} = a^{(i-1)} - a_i q_i \quad (3.9)$$

という計算法もある. 両者は, $q_1 \dots q_m$ が完全に正規直交であり, 計算が無限の精度で行われれば同値である. しかし誤差まで考えると事情が異なる.

もし q_2 が誤差のために q_1 に平行な成分をわずかに含み, $q_2 + \epsilon q_1$ であったとする. (3.8) では a_2 として

$$a_2 = (q_2 + \epsilon q_1)^T a = a_2 + \epsilon a_1 \quad (3.10)$$

が得られ, ϵ に比例する誤差が生じる. 他方 (3.9) では, $a^{(1)}$ との内積を計算するので, a_2 として

$$a_2 = (q_2 + \epsilon q_1)^T (a - a_1 q_1) = a_2 + \epsilon q_1^T a - \epsilon a_1 = a_2 \quad (3.11)$$

となり, ϵ のオーダーの誤差は相殺してしまう. 同様に他の成分についても誤差の相殺が起り精度がよくなる. とくに, 小さい成分の計算にはこの効果が大きい.

具体的には, 行列 A の第 j 列ベクトルを a_j と記して,

$$a_1^{(0)} = a_1 \\ \text{for } j=1 \text{ to } m \text{ do} \\ R_{jj} = \|a_j^{(j-1)}\|, \quad q_j = a_j^{(j-1)} / R_{jj} \quad (3.12) \\ \text{for } k=j+1 \text{ to } m \text{ do} \\ R_{jk} = q_j^T \cdot a_k^{(j-1)}, \quad a_k^{(j)} = a_k^{(j-1)} - q_j R_{jk}$$

により, $Q = (q_1, q_2, \dots, q_m)$ と R とが求められる. 同様に $z = Q^T y$ も, 単に掛けるのではなく,

$$y^{(0)} = y \\ \text{for } j=1 \text{ to } m \text{ do} \quad (3.13) \\ z_j = q_j^T \cdot y^{(j-1)}, \quad y^{(j)} = y^{(j-1)} - q_j z_j$$

によって, q_j に平行な成分を引きながら内積を求める. 最後に残った $y^{(m)}$ は残差である. この方法を修正 Gram-Schmidt 法とよぶ.

3.3 QR 分解の一貫性

A に二通りの QR 分解

$$A = Q_1 R_1 = Q_2 R_2 \quad (3.14)$$

が存在したとする.

$$D \equiv R_1 R_2^{-1} = Q_1^T (Q_2 R_2) R_2^{-1} = Q_1^T Q_2 \quad (3.15)$$

は定義により上三角行列である. 同時にこの転置行列

$$D^T = Q_2^T Q_1 = R_2 R_1^{-1} \quad (3.16)$$

も上三角行列であるから, D は対角行列であり,

$$D^2 = (R_1 R_2^{-1})(R_2 R_1^{-1}) = I \quad (3.17)$$

である. すなわち D は, 対角要素が ± 1 の対角行列である.

$$Q_2 = Q_1 D, \quad R_2 = D R_1 \quad (3.18)$$

であるから, Q の j 列と R の j 行の符号を同時に反

転する自由度を除いて両者は等しい。QR 分解は本質的に一意的である。

3.4 ピボット選択

3.1, 3.2 節では, $\alpha_1, \alpha_2, \dots, \alpha_m$ の順に変換を行うよう説明したが, 丸め誤差の影響を小さくして精度を保つには, 変換の順序が重要である。これを連立方程式解法の用語を借用してピボット選択と呼ぶことにする。ただし要素の選択ではなく, 列の選択である。

しばしば用いられる選択法は, j 回目のステップにおいて $\alpha_k^{(j-1)}$ ($j \leq k \leq m$) のうちノルム最大の列を j 列目と交換する方法である。しかし A の列ベクトルは本来単位も数値的な大きさも異なっている場合が多いので, むしろ変換の途中におけるノルムの相対減少量によって列を選ぶ方がよい。

はじめに各列のノルム $N_j = \|\alpha_j\|$ ($j=1, 2, \dots, m$) を計算しておき, j 回目のステップでは, $k=j, \dots, m$ の中でノルム比 $\|\alpha_k^{(j-1)}\|/N_k$ が最大になる k を求め, k 列と j 列を交換する。この列は, q_1, q_2, \dots, q_{j-1} に平行な成分の割合が最も少なかった列であるから, 新しい軸 q_j を構成するのに最も適した列である。

もし $\|\alpha_k^{(j-1)}\|/N_k$ の最大値がある正数 ϵ_R より小さければ, ランク落ちと判定し, しかるべき処置をとる。 ϵ_R は ϵ_M に近いある数である。このような場合, $\alpha_k^{(j-1)}$ は α_k の丸め誤差と同程度以下であるから, もはや意味のある情報は含まれていないと考えられる。

ランク落ちが起らなかった場合は, 最後のノルム比 $\|\alpha_m^{(m-1)}\|/N_m$ は, A の各列をノルム 1 にスケールした行列 \bar{A} の条件数 $\kappa(\bar{A})$ の下からの評価値を与える²⁾。通常は $\kappa(\bar{A}) \approx \kappa(A)$ であり, 注意深く計算を行えば, 丸め誤差の影響は実質的に $\kappa(\bar{A})$ によって決定されるものと考えられる。

この方法は悪条件に近い場合には威力を発揮するが, 各列のノルムを評価するために計算量が増大する。条件が良いことがあらかじめ分かっている場合には少し簡略化したアルゴリズムが必要であろうと思われる。

3.5 部分倍精度計算

CDC や Cray は別として多くの計算機の ϵ_M は $10^{-6} \sim 10^{-7}$ であるから, 計算の精度を保つためには多くの工夫が必要である。一つの方法はすべてを倍精度で計算することであるが, 演算時間が若干 (2 割位) 増加することはよいとしても, 記憶領域が倍必要になることは大型の問題では致命的である。このような場合, 最終的な結果が単精度でよいならば, 部分倍精度計算

により, ある程度救うことができる。

単精度実数の乗算の結果は本来倍の長さの仮数部を有しているはずであり, 通常は下位を切り捨てて単精度化している。しかし線形計算にしばしば現れる積和計算,

$$S=0.0$$

$$\text{DO } 10 \text{ I}=1, N$$

$$10 \text{ S}=\text{S}+\text{A}(\text{I}) * \text{B}(\text{I})$$

において, もし $\text{A}(\text{I}) * \text{B}(\text{I})$ の結果をすべて生かしたまま倍精度実数 S に加えていければ, 精度を大幅に向上させることができる³⁾。これを部分倍精度演算という。とくに方程式の左辺のように A と B の内積がほとんど 0 になる場合に有効である。

問題はどのようなプログラムを書けば所期の演算を実行してくれるのかということである。実際には, 多くの大型機では, S を倍精度に宣言しておくだけでこの目的は達せられる。しかし JIS 規格ではそうになっていない。最も奇妙なのは, HITAC M200H の FORTRAN 77 において IAP (内蔵アレイプロセッサ) を使う場合である。 S を単精度としておくと, アレイプロセッサの中でちゃんと倍精度で積和を計算してくれるのに, S を倍精度と宣言すると, ベクトル化不能と判定されて効率が落ちてしまう。言語やコンパイラ的设计の際, 数値計算の常識にも十分配慮していただきたい。

もちろん, 部分倍精度計算では, 単精度表現の入力データに元々含まれる誤差まで補正するわけではないが, 演算の安定化に非常に役立つ。

3.6 数値例

アルゴリズムの性能を公平に評価することは簡単ではないが, ここでは Wampler のテストによる結果を示す。Wampler⁶⁾ は, 既存の 20 余種の最小二乗法プログラムを同一の例題についてテストし, 種々の解法の精度を検討した。問題は 5 次式,

$$y_i = \sum_{j=1}^6 x_j q_i^{j-1} \quad (3.19)$$

のあてはめであり, “測定値” は, パラメータの真値

$$\text{(例 1)} \quad x^0 = (1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0)$$

$$\text{(例 2)} \quad x^0 = (1, 0, 0.1, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5})$$

から誤差を加えずに作成した。横座標 q_i は 0, 1, 2, \dots 20 の 21 点である。

この問題は係数行列 A の条件数が 6.4×10^6 であり, 単精度計算では限界に近いケースである。ただしモデル化の点から見れば, 作爲的に条件を悪化させた例で

表-1 線形解法の比較

アルゴリズム	正解桁数							残差二乗和	
	x_1	x_2	x_3	x_4	x_5	x_6	平均		
例 1)	修正 Gram-Schmidt 法	1.9	1.5	1.9	2.8	4.0	5.7	3.0	1.995 E-02
	同 (反復改良)	6.0	*	6.0	*	6.0	*	6.0	2.522 E-04
	Householder 法	0.2	0.2	0.6	1.5	2.8	4.5	1.6	7.751 E-01
	同 (反復改良)	*	*	*	*	*	*	*	0.0
	特異値分解法	0.0	0.0	0.1	0.7	1.8	3.4	1.0	3.655 E+02
	正規方程式 LESM 2	-2.9	-3.2	-2.7	-1.8	0.6	1.1	-1.7	2.482 E+06
同 (倍精度)	7.4	7.1	7.5	8.4	9.7	11.4	8.6	8.505 D-15	
例 2)	修正 Gram-Schmidt 法	6.0	4.5	3.8	3.6	3.8	4.4	4.4	2.205 E-10
	同 (反復改良)	6.0	4.7	4.0	3.8	4.0	4.6	4.5	1.921 E-10
	Householder 法	4.4	3.5	3.1	3.0	3.4	4.1	3.6	3.535 E-09
	同 (反復改良)	6.0	4.7	4.0	3.8	4.0	4.6	4.5	1.921 E-10
	特異値分解法	0.0	0.0	0.1	-1.0	-1.0	-0.4	-0.4	8.838 E+00
	正規方程式 LESM 2	1.7	0.5	0.0	0.1	1.3	0.8	0.5	1.097 E-03
同 (倍精度)	11.5	10.1	9.6	9.4	9.7	10.3	10.1	8.119 D-23	

(* は完全正解を表す)

ある。同じ5次式でも、原点を移動して q_i を $-10 \sim 10$ にシフトすれば、条件数も下り、ずっと計算しやすくなる。また A の各列のスケールが非常に異なっている点も計算に難しさを加えている。各列をノルム1にスケールすれば、条件数は 6.4×10^2 程度である。

この問題を種々のアルゴリズムで解いた結果を表-1に示す。解の精度は正解桁数で示した。正解桁数とは $-\log_{10} \{\max(\epsilon_M, |x_j - x_j^0|/|x_j^0|)\}$ であり、たとえば3.0とは相対誤差が0.1%であることを示す。正規方程式を用いた解法 (LESM 2) は正解桁数が負で、全く誤った値を与えている。QR分解に基づく解法では反復改良³⁾の効果が著しい。真値で残差が0となるためであろう。特異値分解法は意外に精度が悪い。比較のため LESM 2 を倍精度で実行した結果を示したが、例1ではQR分解による反復改良付最小二乗法をわずかに凌ぐ程度である。“測定値”として真値に正規乱数を加えた場合についても実行を行ったが、やはりQR分解に基づく解法がよい結果を与えた。

計算は HITAC M-170 の最適化 FORTRAN を使用し、LESM 2 以外は SALS (単精度版)⁷⁾を用いた。

4. 非線形最小二乗法

モデルがパラメータに対して線形でない場合には非線形最小二乗法

$$S(\mathbf{x}) = \sum_{i=1}^n \{f_i(\mathbf{x})\}^2 = \min \quad (4.1)$$

を解かなければならない。非線形最小二乗法には大別して

a) 一般の最適化法 (シンプレックス法, 巡回緩和

法, Rosenbrock 法, Powell 法, モンテ・カルロ法等々)

b) Newton 系の方法

c) Gauss-Newton 系の方法

の3種に分類できる。a) に属する方法は効率がよくないが、歴史が古く、今でも初期のあらい探索等に用いられることがある。60年代以降c)に属するアルゴリズムが発展してきたが、最近b)の諸方法についても研究が進み、その有効性が実証されてきている⁸⁾。

4.1 Newton 法

この種のアルゴリズムの基本は Newton 法である。 m 元非線形連立方程式

$$g_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, m \quad (4.2)$$

を解くには、 k 番目の近似解 $\mathbf{x}^{(k)}$ から、次の近似値 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \delta \mathbf{x}$ を、一次方程式

$$A \delta \mathbf{x} = -g_i(\mathbf{x}^{(k)}) \quad (4.3)$$

ただし、 $A_{ij} = (\partial g_i(\mathbf{x}) / \partial x_j)_{\mathbf{x}=\mathbf{x}^{(k)}}$ を解いて求める。実際には計算を安定化するために種類の工夫が必要であるが、重根でなければ解のある近傍で二次収束することが知られている。

4.2 Gauss-Newton 法系の方法

優決定系の非線形方程式

$$f_i(\mathbf{x}) = 0 \quad i = 1, 2, \dots, n \quad (n > m) \quad (4.4)$$

の最小二乗解を求める場合、Newton 法と同様に線形化して、優決定系の一次方程式

$$A \delta \mathbf{x} = -f_i(\mathbf{x}^{(k)}) \quad (4.5)$$

ただし、 $A_{ij} = (\partial f_i(\mathbf{x}) / \partial x_j)_{\mathbf{x}=\mathbf{x}^{(k)}}$ を解くことによって反復を行う方法が考えられる。これがいわゆる Gauss-Newton 法である。一回の反復

は線形最小二乗法と全く同じであるので、QR 分解等の高精度解法を用いることもできる。ただし Newton 法と異なり、二次収束の性質はない。

初期値が真の解から大きく離れていたり、 $f(x)$ の非線形性が大きい場合は安定性が悪い。アルゴリズムとしては必ずしも必要とは限らないが、安定化のために、反復の各回において二乗和 $S(x)$ が前回より非増大であることを要求することが多い。正規方程式の係数行列 $A^T A$ は非負定値であるから、少し変形して

$$x^{(k+1)} = x^{(k)} + \alpha \delta x \quad (4.6)$$

とおいて α を十分小さくすれば、

$$S(x^{(k+1)}) \leq S(x^{(k)}) \quad (4.7)$$

とすることができる。 α を小さくすれば安定になるが収束は遅くなり、 α を大きく ≈ 1 にとれば、線形に近い場合には収束が速くなるが不安定性が増大する。 α に関して厳密な極小値一次元探索を行う方法も提案されているが、計算の手間が増えるので、(4.7) 式が成立するまで α を 1/2 倍ずつ減少させていく程度の探索を用いよう。

昔からよく用いられている Marquardt 法⁹⁾ は、Gauss-Newton 法の変形であり、正規方程式の対角要素に付加項を加え

$$(A^T A + \lambda I) \delta x = -A^T f \quad (4.8)$$

によって修正量 δx を計算する。QR 法では、

$$\bar{A} = \begin{bmatrix} A \\ \sqrt{\lambda} I \end{bmatrix}, \quad \bar{f} = \begin{bmatrix} f \\ 0 \end{bmatrix} \quad (4.9)$$

をそれぞれ A, f として適用すればよい。 λ は 0 または正の数であり、0 であれば通常の Gauss-Newton 法に帰差し、 $\lambda \gg \|A^T A\|$ ならば最急降下方向

$$\delta x \approx -\lambda^{-1} A^T f \quad (4.10)$$

となる。すなわち、この方法は両者の折衷である。また λI を加えることにより、(4.8) 式の係数行列の条件数が減少するので、 $A^T A$ が特異または悪条件のとき、反復を安定化するという役割も果している。 λ を十分大きくとれば、丸め誤差を無視できる範囲で(4.7) 式を満たす $x^{(k+1)}$ が必ず見出されるはずである。速くかつ安定に収束させるには λ の選び方が重要であり、種々の提案がなされている。(4.8) 式で δx を求めた上で、この方向に一次元探索を行うというアルゴリズムも提案されている。

4.3 偏微分行列の逐次修正

以上の Gauss-Newton 系のアルゴリズムは、各反復ごとに偏微分行列 A を計算する必要がある。変数の数が多い場合や、式が複雑な場合、誤りなくプログラ

ムすることは至難の業であり、計算時間も多くなる。有限差分によって A の近似値を求める方法もしばしば用いられる。差分のステップ Δx さえ適当であれば有効であるが、 m 回の関数評価がやはり毎回余分に必要になる。Powell の最小二乗法¹⁰⁾ では、最初に一回だけ差分によって A を計算し、それ以後は探索の途中で計算した f の値を用いて A を修正しながら、Gauss-Newton 法を適用する。したがって A のための余分な計算は最初だけである。

近接する 2 点 x と $x + \delta x$ の関数値が、それぞれ f と $f + \delta f$ であったとする。この情報を用いて偏微分行列の近似値 A を改良して A^* とするには、

$$A^* = A + \alpha(\delta f - A \delta x) \cdot \delta x^T / \|\delta x\|^2 \quad 0 < \alpha \leq 1 \quad (4.11)$$

とすればよい。修正項はランク 1 の行列である。 f が線形である場合、この補正によって A より A^* の方が真の偏微分行列 \bar{A} に近づくことを示す。この場合には $\delta f = \bar{A} \delta x$ が成立しているはずであるから、

$$\begin{aligned} A^* \delta x &= A \delta x + \alpha(\bar{A} \delta x - A \delta x) \cdot \delta x^T \delta x / \|\delta x\|^2 \\ &= \alpha \bar{A} \delta x + (1 - \alpha) A \delta x. \end{aligned} \quad (4.12)$$

$\alpha = 1$ の場合には、 δx に平行な成分は正しく補正される。 α は一種の緩和パラメータであり、0.8~1.0 の値とする。 A の誤差をフロベニウス・ノルム $\|M\|_F = \text{Tr}(M^T M)$ で評価すれば、

$$\|A^* - \bar{A}\|_F \leq \|A - \bar{A}\|_F \quad (4.13)$$

が証明される。

Powell の最小二乗法では、上の補正に基づいて、 $(A^T A)^{-1}$ の逐次補正を行っているが、Cholesky 分解

$$A^T A = R^T R \quad (R \text{ は上三角行列}) \quad (4.14)$$

の更新公式¹¹⁾を用いる方が、計算量や誤差の点から有利であろう。

QR 分解 $A = QR$ や、一般逆行列 A^+ の更新公式は筆者の知る限り便利なものはない。それは δf が、 A の列ベクトルの張る空間 $\mathcal{R}(A)$ の外にある可能性があり、そのため Q の再直交化が必要になるからである。

4.4 Powell のハイブリッド法

Powell の最小二乗法¹⁰⁾は、偏微分行列 A の逐次補正を用いているので、補正が効率よく機能するには、過去 m 回の修正ベクトル δx が互いに直交に近いことが必要である。一度ある方向について二乗和の減少が少ないと、以後その方向に δx がとられにくくなり、したがってその方向に関する偏微分がいつまでも更新されず、事実上その方向は死んでしまう。そのた

め Powell の最小二乗法は、多くの非線形問題に対して優れた性能を有しているにもかかわらず、場合によっては部分空間の中に探索方向が限られてしまうという欠点指摘されていた。

Powell のハイブリッド法¹²⁾では、この点を次のように改良した。Powell は、過去 $2m$ 回の修正ベクトルが、常にパラメータ空間 (m 次元) 内の全方向に満遍なく広がっていること、すなわち、空間内の任意のベクトルに対し、これと 60° 以内 (または 120° 以上) の角を成す修正ベクトルが $2m$ 個の中に少なくとも 1 個は存在することを保証する巧妙なアルゴリズムを提案した。もしこの条件が破れそうときには、偏微分行列の更新のためだけに f の評価を行う。

ハイブリッド法のもう一つの特徴は、修正ベクトルの長さ $\|\delta x\|$ に上限 d を設定し、Gauss-Newton 法の解 δx_0 の長さが d を越えた場合には、最急降下法の解 δx_0 との線形結合で長さが d となるよう δx を決定することである。 $m=2$ の場合には、このアルゴリズムは Marquardt 法と同様に振る舞うことが示される¹³⁾。 d の大きさは、adaptive に調節される。Powell 自身によると、ハイブリッド法は、昔の Powell の最小二乗法を完全に凌駕しているという¹²⁾。

4.5 Newton 系の方法

(4.1) 式の二乗和 $S(x)$ の最小点 x は、連立方程式

$$\frac{\partial S(x)}{\partial x_j} = 2 \sum_{i=1}^n f_i(x) \frac{\partial f_i(x)}{\partial x_j} = 0, \quad j=1, 2, \dots, m \quad (4.15)$$

を満たす。この方程式の解が極小点になっているかどうかは、二階以上の偏微分によって決まる。方程式 (4.15) を Newton 法で解くには、一次方程式

$$\sum_i \left(\sum_j A_{ij} A_{ij} + \sum_i f_i \frac{\partial^2 f_i}{\partial x_j \partial x_i} \right) \delta x_i = - \sum_i A_{ij} f_j(x^{(k)}) \quad (4.16)$$

によって δx を計算すればよい。左辺の括弧内を (j, l) 要素とする $m \times m$ 対称行列 H は、 $S(x)/2$ のヘシアン $(\partial^2 S / \partial x_j \partial x_i) / 2$ である。Newton 法は、 H が極小点付近で正定値であれば (すなわち重根でなければ) 解の近傍で二次収束する。したがって、よい近似値を初期値に選べば収束は非常に速い。しかし収束領域が必ずしも広くないのが欠点である。もう一つの欠点は、反復の各ステップで f の二階偏微分の計算が必要なことであり、このままでは実用的ではない。前述の Gauss-Newton 法は、ヘシアンの第 2 項の二階偏微分を無視したものとみることでもでき、計算量が少なくて

すむかわりに二次収束しない。

4.3 節において、 f の値を用いて一階偏微分 A を逐次修正する方法について述べたが、同様に、 f と A のみを直接計算し、これらの値によって H を逐次修正することができる。このようなアルゴリズムを用いた最小化法を一般に quasi-Newton 法とよぶ。同時に、収束の安定性を高めるために、ゆるい直線探索を併用する。quasi-Newton 法にはさまざまな変種があり、田辺の総合報告⁹⁾に詳しく解説されている。田辺は、数値実験の結果、SSR (対称かつランク 1) 公式に基づいた (修正) Biggs 法が総合的に優れた性能を持つという結論を得ている。

この方法では、 H の第 1 項は $A^T A$ で与え、第 2 項

$$S(x)_{j1} = \sum_{i=1}^n f_i(x) \frac{\partial^2 f_i(x)}{\partial x_j \partial x_i} \quad (4.17)$$

を逐次修正する。2 点 x_1, x_2 で一階偏微分行列 A_1, A_2 が求められていたとする。 f の二階偏微分の変化を無視すれば、

$$S(x_2)(x_2 - x_1) = (A_2 - A_1)^T f(x_2) \quad (4.18)$$

が成立するはずであるから、この式が成立するように、 S の修正値 S^* を求める。対称性を保つランク 1 の補正項を $s \cdot s^T$ と仮定すれば、

$$S^*(x_2) \delta x = \delta A^T \cdot f_2 \quad (4.19)$$

ただし、 $\delta x = x_2 - x_1$, $\delta A = A_2 - A_1$, $f_2 = f(x_2)$

より、

$$S^*(x_2) = S + \frac{(\delta A^T f_2 - S \delta x) \cdot (\delta A^T f_2 - S \delta x)^T}{(\delta A^T f_2 - S \delta x)^T \cdot \delta x} \quad (4.20)$$

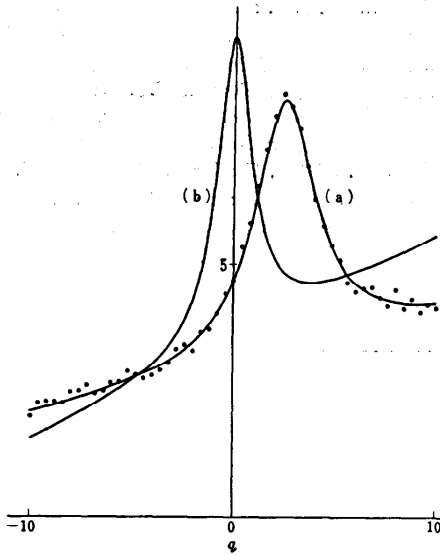
を得る。田辺は、このアルゴリズムを、NOLLS という非線形最小二乗法サブルーチン・パッケージの上に実現した。なお NOLLS では、(4.20) 式の分母が小さくなりすぎた場合には BFGS 公式⁹⁾に切り換えている。

4.6 計算例

一例として、ピークのあてはめ問題を各解法によって解いた例を示す。モデル関数は

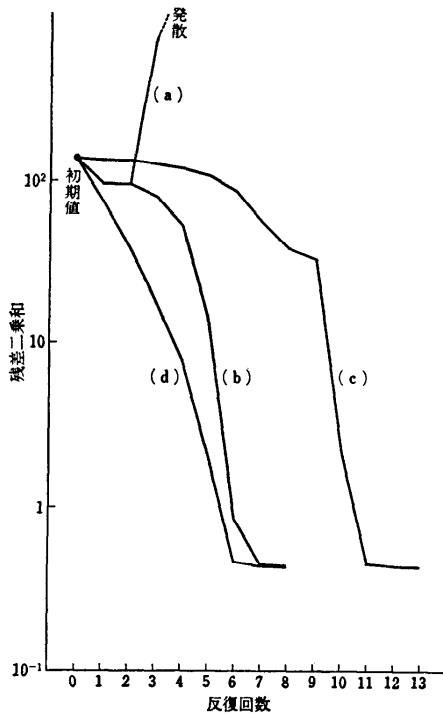
$$f(q) = \frac{h \Gamma^2}{(q - q_0)^2 + \Gamma^2} + a_0 + a_1 q \quad (4.21)$$

で、直線のバックグラウンドの上に Lorentz 型のピークが乗っている (図-1)。51 点のデータから 5 個のパラメータを決定する問題である。解法の性能を調べるために真値からかなり離れた値を初期値とした。各解法の二乗和 $S(x^{(k)})$ の振舞いを図-2 に示す。



(a)は真のモデル, (b)はパラメータの初期値に対応するモデル. 点は正規乱数を加えて作成した“測定値”.

図-1 ピークのあてはめ



(a) Gauss-Newton 法, (b) Marquardt 法,
(c) ハイブリッド法, (d) NOLLS (Biggs 法)

図-2 各解法の残差二乗和の振舞

Gauss-Newton 法では収束していない。ハイブリッド法では、初期での S の減少が遅く回数を損している。今後、種々のテストにより各アルゴリズムの特徴を明らかにしていかなければならない。

5. む す び

以上最小二乗法のアルゴリズムを概観したが、いくら良いアルゴリズムが発見されても、利用可能なプログラムの形になっていなければ画に書いた餅である。今だに、正規方程式を Gauss-Jordan 法で解いたり、最急降下法やシンプレックス法のみを頼っている例が少なくない。アルゴリズムの開発だけでなく、その社会的普及が大きな課題である。

この小文を準備するにあたり、中川徹、田辺國士、戸川隼人 3 氏の助言を感謝いたします。

参 考 文 献

- 1) Forsythe, G. E., Malcolm, M. A. and Moler, C. B.: *Computer Methods for Mathematical Computations*, Prentice-Hall (1977). 森正武訳: 計算機のための数値計算法, 科学技術出版社 (1978).
- 2) 松井正一, 伊理正夫: あふれのない浮動小数点表示方式, 情報処理学会論文誌, Vol. 21, No. 4, pp. 306-313 (1980). 浜田穂積: 二重指数分割に基づくデータ長独立実数値表現法, 情報処理学会論文誌, Vol. 22, No. 6, pp. 521-526 (1981).
- 3) Forsythe, G. E. and Moler, C. B.: *Computer Solution of Linear Algebraic Systems*, Prentice Hall (1967). 渋谷政昭, 田辺國士訳: 計算機のための線形計算の基礎, 培風館 (1969).
- 4) Lawson, C. L. and Hanson, R. J.: *Solving Least Squares Problems*, Prentice-Hall (1974).
- 5) Björk, Å.: *Solving Linear Least Squares Problem by Gram-Schmidt Orthogonalization*, BIT, Vol. 7, pp. 1-21 (1967). 渋谷政昭: 最小二乗法のアルゴリズム, 応用統計学, Vol. 1, No. 1, pp. 3-16 (1971).
- 6) Wampler, R. H.: *Proceedings of the Computer Science and Statistics Sixth Annual Symposium on the Interface*, p. 94 (1972).
- 7) 中川 徹, 小柳義夫: 最小二乗法標準プログラム SALS (第2版) 利用の手引, 東京大学大型計算機センター (1979).
- 8) 田辺國士: 非線形最小二乗法のアルゴリズム, 応用統計学, Vol. 9, No. 3, pp. 119-140 (1981).
- 9) Levenberg, K.: *A Method for the Solution of Certain Nonlinear Problems in Least Squares*, Quart. Appl. Math., Vol. 2, pp. 164-168 (1944).

- Marquardt, D. W.: *An Algorithm for Least Squares Estimation of Nonlinear Parameters*, SIAM J., Vol. 11, pp. 431-441 (1963).
- 10) Powell, M. J. D.: *A Method for Minimizing a Sum of Squares of Nonlinear Functions without Calculating Derivatives*, Comp. J., Vol. 7, pp. 303-307 (1965).
- 11) Gill, P. E. and Murray, W.: *Quasi-Newton Methods for Unconstrained Minimization*, J. Inst. Math. Appl., Vol. 9, pp. 91-108 (1972).
- 12) Powell, M. J. D.: *AERE Harwell Library Subroutine VA 05 A*, Harwell, Oxfordshire, England (1969). この中の、一般逆行列の更新公式には問題がある。Powell, M. J. D.: *A Hybrid Method for Nonlinear Algebraic Equations*, pp. 87-114, Gordon & Breach (1970).
- 13) 中川 徹, 小柳義夫: 最小二乗法による実験データの解析, 東大出版会 (1982).
(昭和 56 年 10 月 12 日受付)
-